УДК 621.865.8 DOI: 10.18698/2541-8009-2018-02-252

ИССЛЕДОВАНИЕ АЛГОРИТМОВ ДЕТЕКТИРОВАНИЯ ОБЪЕКТОВ ДЛЯ СИСТЕМЫ ТЕХНИЧЕСКОГО ЗРЕНИЯ ПОЖАРНОГО РОБОТА

В.В. Попов

v1.po@ya.ru
SPIN-код:1324-8412

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Аннотация

Проанализированы современные интеллектуальные алгоритмы распознавания образов с целью выбора наиболее оптимального для последующей реализации в системе технического зрения робота-пожарного. Использованы методы поиска ближайшего соседа, линейной классификации и нейронных сетей. Приведены основные принципы, лежащие в основе работы каждого метода, а также их достоинства и недостатки. Выбран наиболее совершенный по совокупности показателей качества работы алгоритмов метод, следовательно, наиболее подходящий для использования в системах технического зрения.

Ключевые слова

Техническое зрение, распознавание образов, обработка изображений, функция потерь, алгоритм линейной классификации, алгоритм поиска ближайшейго соседа, нейронные сети

Поступила в редакцию 04.12.2017

Введение. Значительное упрощение работы пожарного расчета возможно благодаря использованию современной техники, в число которой входят пожарно-спасательные роботы (рис. 1). Ими все более широко оснащают мобильные пожарные бригады, различные промышленные и культурно-массовые объекты.

Рис. 1. Мобильный робототехнический комплекс разведки и пожаротушения

Современные разработчики робототехники в отрасли пожарного дела уверены [1], что приоритет в этой отрасли за интеллектуальными саморегулирующимися системами. Это обусловлено высоким уровнем техногенного раз-

вития производства, вследствие чего возможны различные катастрофы и ЧС. Ряд моделей пожарных роботов уже проводит круглосуточный мониторинг на объекте и при возгорании немедленно ликвидируют пламя струей огнегасящего вещества.

Для обеспечения автономной работы робота-пожарного его необходимо оснастить системой технического зрения (СТЗ), реализующей алгоритмы детектирования очагов возгорания.

Цель данной работы — проанализировать современные интеллектуальные методы детектирования объектов, а также выявить их основные преимущества и недостатки.

Перед СТЗ робота-пожарного ставится задача эффективного определения очагов возгорания на основе анализа входящего видеопотока. Судить о наличии пожара на изображении можно, анализируя такие признаки, как дым и оттенки белого, желтого и красного цветов, образующие некоторые геометрические фигуры (языки пламени).

Предполагается, что робот будет работать как внутри зданий, так и снаружи при различных погодных условиях. Руководствуясь этими соображениями, будем требовать хорошей степени инвариантности алгоритма к условиям освещения, в которых будет работать робот-пожарный.

Поскольку СТЗ будет установлена на робота, представляющего собой мобильную платформу, к которой предъявляются требования автономности, то подходящий метод не должен требовать больших вычислительных мощностей, то есть значительного энергопотребления. Кроме того, в силу неоднородности рельефа, где робот должен успешно решать поставленные задачи, очаг возгорания может располагаться на любом возвышении и под любым наклоном по отношению к роботу. Для корректной работы СТЗ в этом случае необходимо, чтобы реализуемый алгоритм был невосприимчив к изменению ориентации распознаваемого объекта в пространстве.

Распознавание образов. Современные методы распознавания образов на изображении состоят из следующих этапов [2]:

1) загрузка в память вычислительного устройства набора изображений, разделенных на классы (рис. 2);

2) обработка и сопоставление анализируемого изображения с набором предварительно загруженных классифицированных изображений;

3) генерирование вычислительным устройством предсказания о классах объектов на анализируемом изображении, основываясь на результатах работы этапа 2.

В зависимости от способов реализации того или иного шага, наиболее популярными считают алгоритмы (методы):

– поиска ближайшего соседа;

– поиска K ближайших соседей;

– линейной классификации;

– нейронных сетей.

Рассмотрим каждый из перечисленных методов подробнее.

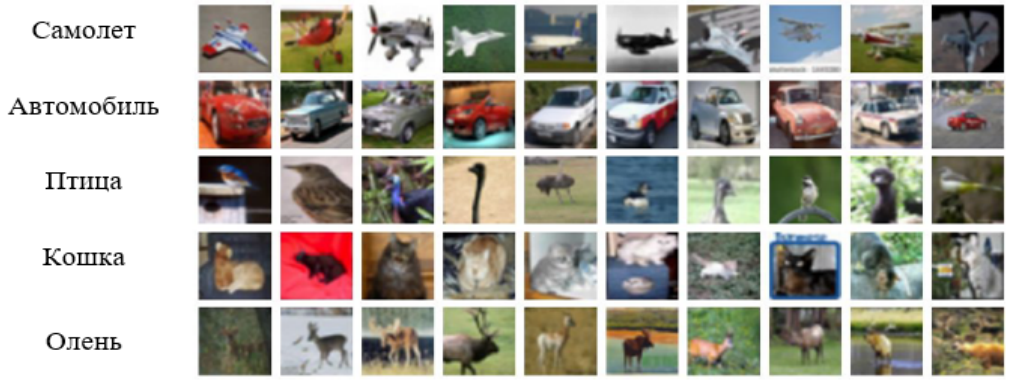

Рис. 2. Набор классифицированных изображений

Алгоритм поиска ближайшего соседа. Данный алгоритм позволяет классифицировать объект на изображении, сравнивая его с набором предварительно загруженных изображений, разделенных на классы.

Каждое изображение представляет собой матрицу пикселей, каждый элемент которой является в общем случае трехмерным вектором компонент RGB. Из этой трехмерной матрицы получают двумерную, заменяя каждый пиксель его значением в оттенках серого. Затем из набора предварительно загруженных изображений выбирается наиболее похожее на анализируемое. Это происходит в результате вычитания из матрицы соответствующей анализируемому изображению матрицы сравниваемого изображения. Полученная разность берется по модулю, а значения всех элементов матрицы суммируются (рис. 3). Чем меньше число получится в результате, тем более схожими являются два изображения. Алгоритм на выходе определяет класс, соответствующий предварительно загруженному изображению, наиболее похожему на анализируемое.

Рис. 3. Алгоритм поиска ближайшего соседа для изображений размером 4×4 пикселя

Для повышения точности работы алгоритма была создана его более совершенная версия, получившая название ***алгоритма поиска K ближайших соседей*** [3]. Единственным отличием в работе этих алгоритмов является поиск не од-

ного минимального результата сравнения (см. рис. 3), а K таких результатов, из анализа которых определяется искомый класс путем простого большинства. Заключительный этап работы модифицированного метода представлен на рис. 4.

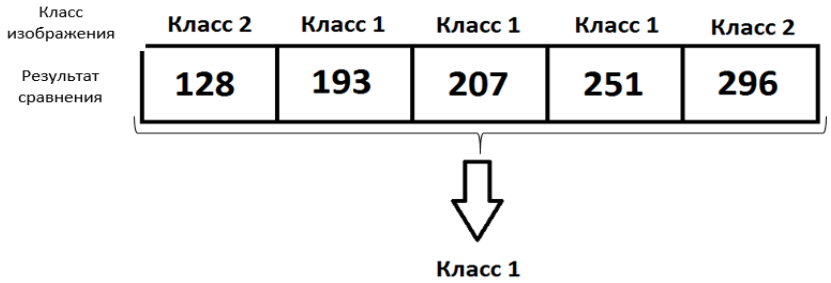

Рис. 4. Заключительный этап работы алгоритма поиска K ближайших соседей

Алгоритм поиска ближайших соседей, как и его модифицированная версия, достаточно редко применяются в задачах классификации изображений, поскольку им присущи серьезные недостатки:

1) требуется очень большое количество предварительно загруженных классифицированных изображений для корректной работы алгоритма;

2) величина разности пикселей не всегда правильно отображает различия между изображениями, из которых они составлены;

3) низкая скорость работы.

Алгоритм линейной классификации. Данный метод также известен как алгоритм параметрической классификации [4]. Рассмотрим его работу на примере, представленном на рис. 5. Допустим, нам требуется классифицировать изображение в оттенках серого размером 2×2 пикселя. Это матрица пикселей преобразуется в вектор–столбец **x**, который затем умножается на весовую матрицу **W**, а к полученному значению добавляется вектор **b**, определяющий потенциальную повышенную вероятность принадлежности исследуемого изображения к тому или иному классу. После проведения этих операций на выходе получается вектор **f**, каждый элемент которого определяет вероятность принадлежности анализируемого изображения к соответствующему классу. При этом большие значения элементов выходного вектора соответствуют большей вероятности.

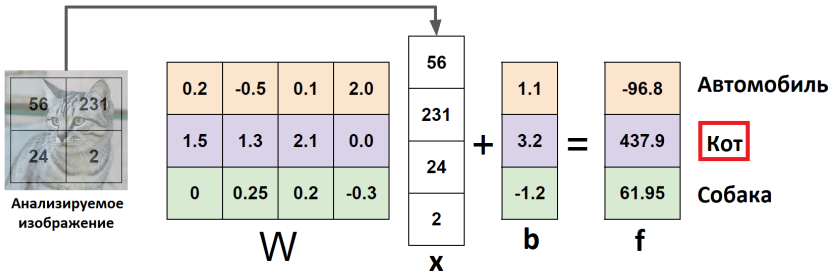

Рис. 5. Алгоритм работы метода линейной классификации

Данный метод значительно быстрее, по сравнению с алгоритмом поиска ближайшего соседа, так как при его работе не происходит поочередного сравнения анализируемого изображения со всеми предварительно загруженными. Вместо этого вычисляется всего один результат перемножения матриц.

Существенным недостатком алгоритма линейной классификации является то, что после формирования матрицы **W** каждому классу соответствует постоянная строка параметров (см. рис. 5), образующих только один шаблон для каждого класса [5].

Самым трудоемким в этом методе является подбор элементов весовой матрицы **W**. Для поиска оптимальной матрицы **W** необходим механизм сравнения весовых матриц. С этой целью используют так называемую функцию потерь **L**, после чего приступают к поиску матрицы **W**, то есть оптимизации весовой функции. Рассмотрим подробнее каждый из этих элементов поиска матрицы **W**.

Функция потерь. Как было сказано выше, функция потерь показывает, насколько верно разработанный алгоритм определил принадлежность анализируемого изображения к тому или иному классу [6]. Функция потерь для матрицы **W** вычисляется как среднее арифметическое значений функции потерь для выходных векторов **f**, полученных в результате обработки каждого из *N* изображений:

$$L = \frac{1}{N}\sum_i L_i\left[\mathbf{f}\left(x_i, \mathbf{W}\right), y_i\right],$$

где y_i — номер правильного класса для *i*-го изображения.

Существуют две наиболее популярные модели представления функции потерь — мультиклассовый метод опорных векторов и метод Софтмакс. Эти методы различаются способами определения функции потерь L_i для результатов исследования конкретного изображения. Рассмотрим каждый из методов отдельно.

Мультиклассовый метод опорных векторов. В данном случае функция потерь L_i вычисляется по формуле:

$$L_i = \sum_{j\neq y_i}\begin{cases}0, \text{ если } s_{y_i} \geq s_j + 1\\ s_j + 1 - s_{y_i}, \text{ если } s_{y_i} < s_j + 1\end{cases} = \sum_{j\neq y_i}\max(0,\ s_j + 1 - s_{y_i}),$$

где S_k — значение элемента вектора **f** для класса *k*.

Из этой формулы следует, что потери будут нулевыми в случае, когда значение элемента вектора **f** для правильного класса будет превышать значения остальных элементов вектора **f** не менее чем на величину запаса, принятую равной единице (рис. 6).

$L_1 = \max(0; 5{,}1 + 1 - 3{,}2) + \max(0;\ -1{,}7 + 1 - 3{,}2) = 2{,}9 + 0 = 2{,}9;$

$L_2 = \max(0; 1{,}3 + 1 - 4{,}9) + \max(0; 2{,}0 + 1 - 4{,}9) = 0 + 0 = 0;$

$L_3 = \max(0; 2{,}2 + 1 - (-3{,}1)) + \max(0; 2{,}5 + 1 - (-3{,}1)) = 6{,}3 + 6{,}6 = 12{,}9;$

$L = (L_1 + L_2 + L_3) / 3 = 5{,}27.$

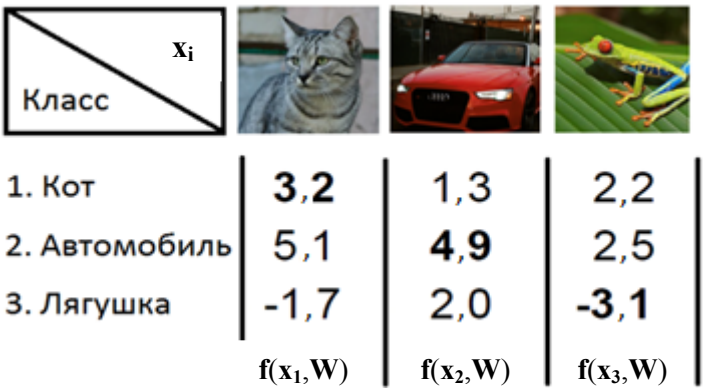

Рис. 6. Входные данные для вычисления функции потерь

Основным недостатком данного метода является то, что он совершенно не реагирует на изменения значений элементов вектора **f** в случае, когда разность между этими элементами и элементом, соответствующим правильному классу, имеет такое же отношение с единицей, как и до изменения [4].

Алгоритм Софтмакс. В данном случае функция потерь L_i вычисляется по формуле:

$$L_i = -\log \frac{e^{s_{y_i}}}{\sum_j e^{s_j}}.$$

Найдем значение функции потерь для изображения, например, кота (рис. 7):

$$L_1 = -\log \frac{e^{3,2}}{e^{3,2} + e^{5,1} + e^{-1,7}} = 0,89.$$

Стоит отметить, что под знаком логарифма находится нормализованное значение элемента вектора $\mathbf{f}_1$, обозначающее вероятность изображения кота.

Рис. 7. Алгоритм Софтмакс

Данный метод, в отличие от мультиклассового метода опорных векторов, является чувствительным к любым изменениям значений элементов вектора **f** [7], что, несомненно, является большим преимуществом, так как позволяет точнее настраивать весовую матрицу **W**.

Оптимизация весовой функции. Процесс оптимизации весовой матрицы **W** заключается в последовательном переходе от одного значения параметров **W** к другому, обеспечивающему уменьшение функции потерь L [8]. При многократном повторении таких переходов удается уменьшить L до значений близких к нулю. Основным методом, позволяющим провести описанные выше операции, является метод градиентного спуска [6], который определяет матрицу **W**, соответствующую локальному минимуму функции L.

Градиент представляет собой вектор частный производных функции и указывает на направление наиболее быстрого роста функции. Поскольку в нашем случае требуется решить задачу минимизации, то рассматривается отрицательный градиент, показывающий направление скорейшего убывания функции и называемый антиградиентом.

На первом этапе метода вычисляется градиент функции L в точке θ^n :

$$\nabla L(\theta^n) = \frac{\delta L(\theta)}{\delta \theta_1^n} + \frac{\delta L(\theta)}{\delta \theta_2^n} + ... + \frac{\delta L(\theta)}{\delta \theta_k^n}.$$

Далее осуществляется изменение вектора параметров **x** в направлении антиградиента с учетом установленного шага h:

$$\theta_i^{n+1} = \theta_i^n - h\frac{\delta L(\theta^n)}{\delta \theta_i^n}.$$

Указанные выше операции повторяются до тех пор, пока значение градиента функции потерь L не окажется меньше установленного значения. В результате получаем вектор параметров θ, заполнение значениями которого весовой матрицы **W** будет соответствовать минимальному значению функции потерь L.

Нейронные сети. Как отмечалось выше, основным недостатком алгоритма линейной классификации является то, что используемая в нем весовая матрица **W** создает только один паттерн (набор весовых коэффициентов) для каждого класса объектов. Это может приводить к некорректной работе алгоритма распознавания образа, например в случае, когда система может корректно распознать объект зеленого цвета, но не способна справиться с задачей, когда этот же объект имеет другой цвет. Для преодоления подобных трудностей и повышения точности работы системы классификации были созданы нейронные сети, которые представляют собой несколько последовательно соединенных слоев [5], каждый из которых является линейным классификатором (рис. 8).

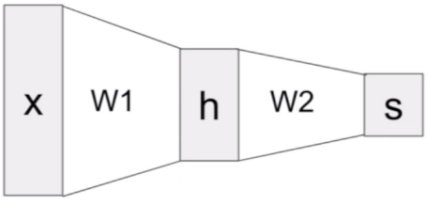

Рис. 8. Архитектура двухслойной нейронной сети

Вектор **x** называют входным слоем, *h* — промежуточным или скрытым, *s* — выходным слоем. Важно отметить, что для эффективной работы нейронной сети необходимо, чтобы хотя бы на одном из ее слоев происходило некоторое нелинейное преобразование координат соответствующего вектора. В противном случае, то есть когда нейронная сеть будет использовать только линейные преобразования, нейронная сеть схлопнется в один линейный слой и по эффективности будет мало превосходить алгоритм линейной классификации.

В настоящее время существует множество разновидностей нейронных сетей. Для задач классификации изображений наилучшим образом себя зарекомендовала архитектура под названием сверточная нейронная сеть [9].

Сверточные нейронные сети. В сверточной нейронной сети не происходит операции скалярного произведения всех пикселей входного изображения **x** на весовую матрицу **W**. Вместо этого вводится матрица-фильтр **F**, размер которой меньше размера **W** [10]. Исходное изображение (матрица **x**) разбивается на области с таким же размером, как у **F**. Затем происходит перемножение матрицы-фильтра на каждую из этих областей. В результате формируется матрица, называемая активационной картой (рис. 9).

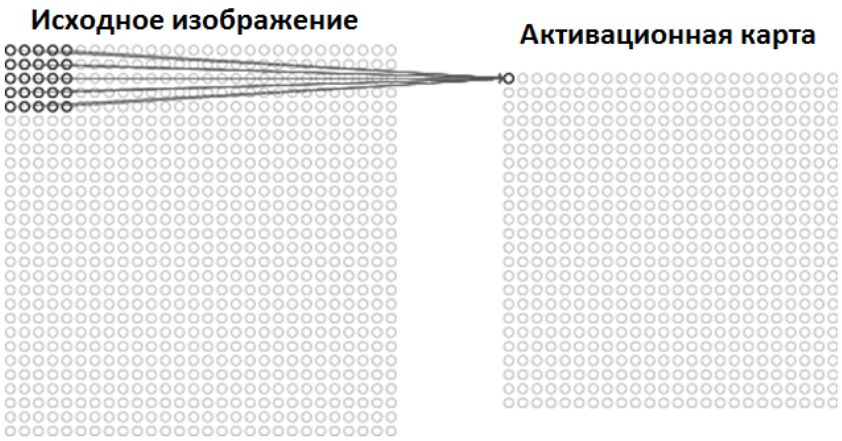

Рис. 9. Формирование активационной карты

Активационных карт может быть несколько, что достигается применением нескольких фильтров **F** к исходному изображению. В результате формируется набор свойств объекта (заключенных в полученных матрицах активационных карт), что позволяет эффективнее детектировать объекты на изображении.

Размер матрицы активационной карты меньше размера матрицы исходного изображения. Этот факт дает возможность использовать сверточные нейронные сети для обработки изображений больших размеров практически без увеличения времени алгоритма классификации, так как применение нескольких слоев в сверточной нейронной сети позволяет значительно уменьшить размеры матрицы пикселей исходного изображения.

Выводы. Показано, что сверточные нейронные сети в настоящее время являются наиболее эффективным методом интеллектуального распознавания объектов благодаря реализации принципов функционирования алгоритма линейной классификации. Помимо эффективности, преимуществами данного метода считают быстродействие и отсутствие требований к размерам входных изображений.

Литература

[1] Вазаев А.В., Носков В.П., Рубцов И.В., Цариченко С.Г. Комплексированная СТЗ в системе управления пожарного робота. *Известия ЮФУ. Технические науки*, 2017, № 1(126), с. 121–132.
[2] Зеленцов И.А. *Распознавание образов*. Москва, Изд-во МГТУ им. Н.Э. Баумана, 2008, 17 с.
[3] Forsyth D.A., Ponce J. *Computer vision: a modern approach*. Pearson, 2011, 792 p.
[4] Simon J.D. Prince. *Computer vision: models, learning and inference*. Cambridge University Press, 2012, 598 p.
[5] Szeliski R. *Computer vision: algorithms and applications*. Springer, 2010, 812 p.
[6] Каллан Р. *Основные концепции нейронных сетей*. Москва, Вильямс, 2001, 287 с.
[7] Menshawy A., Karim Md.R., Zaccone G. *Deep learning with TensorFlow*: *explore neural networks with Python*. Packt Publishing, 2017, 320 p.
[8] Жуков Л.А., Решетникова Н.В. *Учебное пособие по дисциплине «Приложения нейронных сетей»*. Красноярск, Политехнический институт СФУ, 2007, 154 с.
[9] Хайкин С. *Нейронные сети. Полный курс*. Москва, Вильямс, 2006, 1104 p.
[10] Goodfellow I., Bengio Y., Courville A. *Deep learning*. The MIT Press, 2016, 800 p.

Попов Виктор Владимирович — студент кафедры «Робототехнические системы и мехатроника», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Научный руководитель — Рубцов Василий Иванович, кандидат технических наук, доцент кафедры «Робототехнические системы и мехатроника», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

SOFTWARE-DEFINED NETWORKING CONCEPT IMPLEMENTATION TECHNOLOGIES

V.V. Popov v1.po@ya.ru
SPIN-code:1324-8412

Bauman Moscow State Technical University, Moscow, Russian Federation

Abstract

The article analyses current intelligent image recognition algorithms with the view of selecting the most appropriate one for subsequent implementation in the computer vision system of the fire extinguishing robot. We use the methods of searching the nearest neighbor, linear classification and neural networks. The main principles of each method as well as their benefits and drawbacks are considered. We choose a method which is the most advanced one according to the combination of algorithms performance indices and, consequently, the most suitable one for being applied in the computer vision systems.

Keywords

Computer vision, image recognition, image processing, loss function, linear classification algorithm, searching the nearest neighbor algorithm, neural networks

References

[1] Vazaev A.V., Noskov V.P., Rubtsov I.V., Tsarichenko S.G. Combined computer vision system in firefighting robot control system. *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2017, no. 1(126), pp. 121–132.
[2] Zelentsov I.A. Raspoznavanie obrazov [Pattern recognition]. Moscow, Bauman Press, 2008, 17 p.
[3] Forsyth D.A., Ponce J. Computer vision: a modern approach. Pearson, 2011, 792 p.
[4] Simon J.D. Prince. Computer vision: models, learning and inference. Cambridge University Press, 2012, 598 p.
[5] Szeliski R. Computer vision: algorithms and applications. Springer, 2010, 812 p.
[6] Callan R. The essence of neural networks. Prentice Hall, 1998, 248 p. (Russ. ed.: Osnovnye kontseptsii neyronnykh setey. Moscow, Vil'yams publ., 2001, 287 p.)
[7] Menshawy A., Karim Md.R., Zaccone G. Deep learning with TensorFlow: explore neural networks with Python. Packt Publishing, 2017, 320 p.
[8] Zhukov L.A., Reshetnikova N.V. Uchebnoe posobie po distsipline «Prilozheniya neyronnykh setey» [Study guide on "Neural networks application" course]. Krasnoyarsk, Politekhnicheskiy institut SFU publ., 2007, 154 p.
[9] Haykin S.S. Neural networks: a comprehensive foundation. Macmillan, 1994, 696 p. (Russ. ed.: Neyronnye seti. Polnyy kurs. Moscow, Vil'yams publ., 2006, 1104 p.)
[10] Goodfellow I., Bengio Y., Courville A. Deep learning. The MIT Press, 2016, 800 p.

Popov V.V. — student, Department of Robotics and Mechatronics, Bauman Moscow State Technical University, Moscow, Russian Federation.

Scientific advisor — Rubtsov V.I., Cand. Sc. (Eng.), Assoc. Professor, Department of Robotics and Mechatronics, Bauman Moscow State Technical University, Moscow, Russian Federation.